

BlackHoles@Home

Status Report

Zach Etienne



NRPy+: Python-based C code generation framework for NR

*Tensorial
expressions in
Einstein-like
notation ⇨
Highly optimized
C-code kernels
(with FDs)*



"Nerpy", the NRPy+ mascot. Photo CC2.0 [Pacific Environment](#) (modified).

<https://nrpyplus.net>

<https://github.com/zachetienne/nrpytutorial>

BlackHoles@Home

<https://blackholesathome.net>

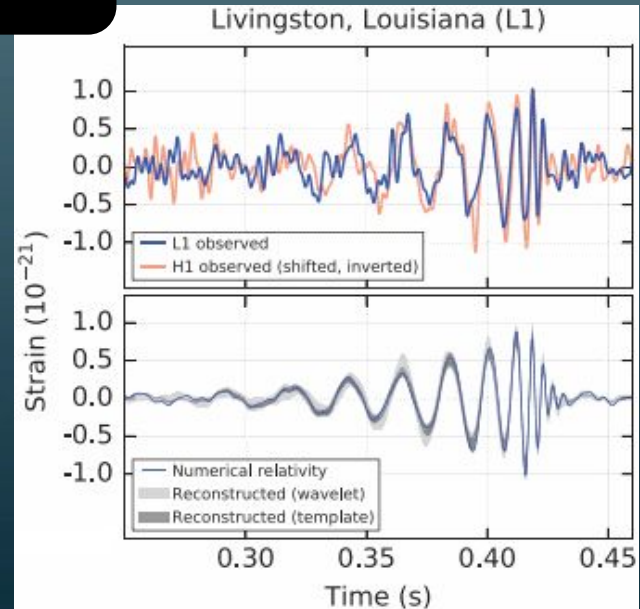
BlackHoles@Home: A proposed volunteer computing project to fit numerical relativity BBH simulations on consumer-grade desktop computers.

Goals: GW catalog generation & follow-ups at scales needed for e.g., LISA.

BlackHoles@Home

Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

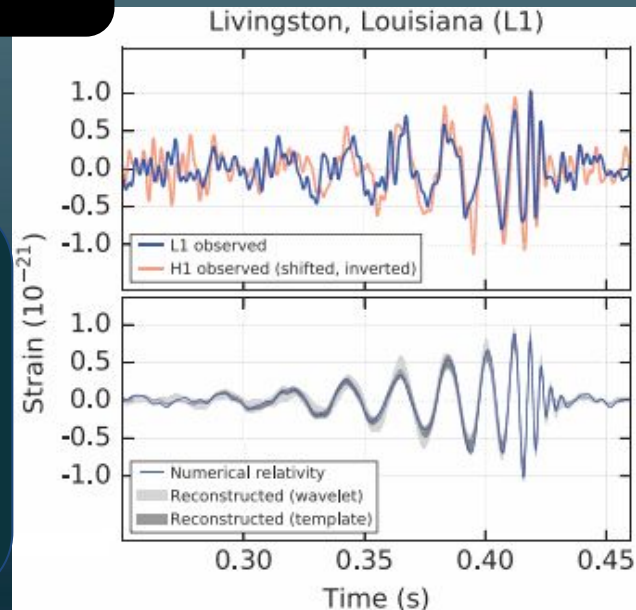


BlackHoles@Home

Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

- Inferring source properties from grav. waves *tough*
 - Binary black holes: **7 dims of parameters!**
 - Current NR grav. wave catalogs:
 - ~3,000 waveforms/15 years
(~3 points per dimension)
 - ***Not enough*** moving forward!
 - Need to grow these catalogs!

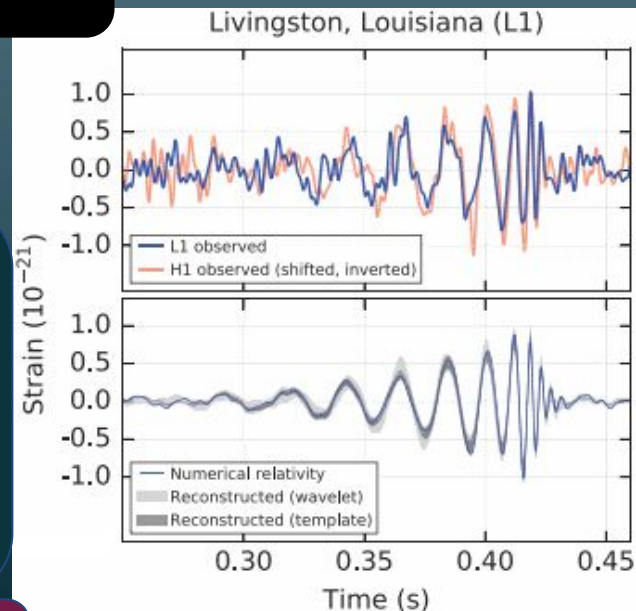


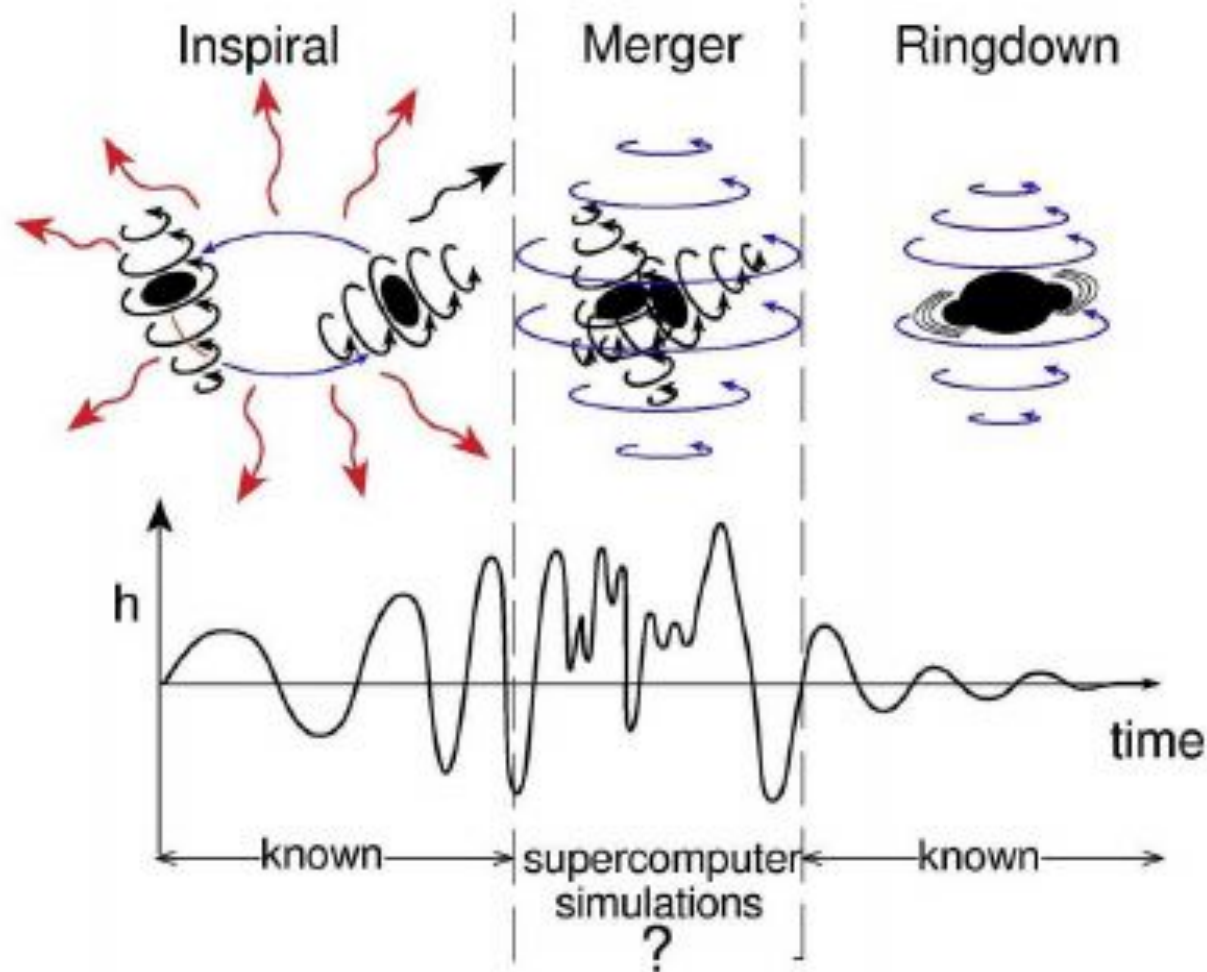
BlackHoles@Home

Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

- Inferring source properties from grav. waves *tough*
 - Binary black holes: **7 dims of parameters!**
 - Current NR grav. wave catalogs:
 - ~3,000 waveforms/15 years
(~3 points per dimension)
 - ***Not enough*** moving forward!
 - Need to grow these catalogs!
- Problem: BBH sims need ~4 supercomputing nodes





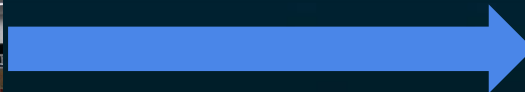
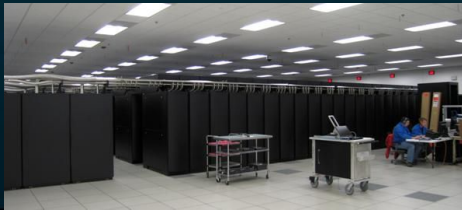
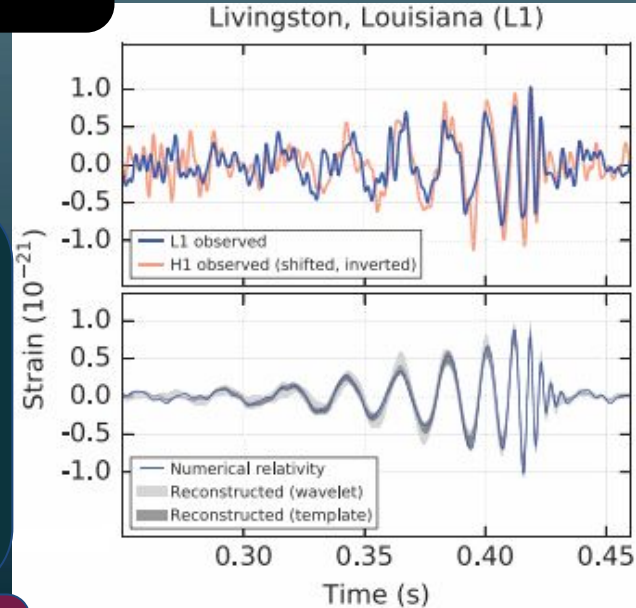
Kip Thorne, pre-2004

BlackHoles@Home

Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

- Inferring source properties from grav. waves *tough*
 - Binary black holes: **7 dims of parameters!**
 - Current NR grav. wave catalogs:
 - ~3,000 waveforms/15 years
(~3 points per dimension)
 - ***Not enough*** moving forward!
 - Need to grow these catalogs!
- Problem: BBH sims need ~4 supercomputing nodes
- Idea: Fit BBH on desktop \Rightarrow massively grow catalogs!

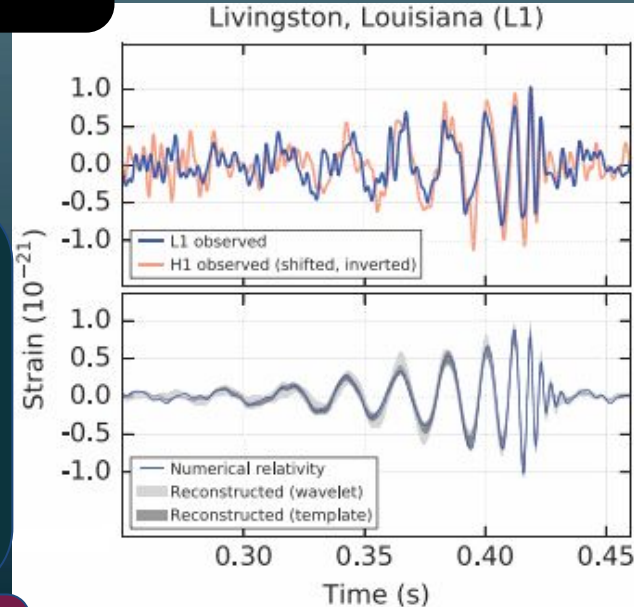


BlackHoles@Home

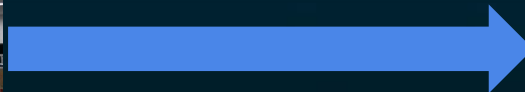
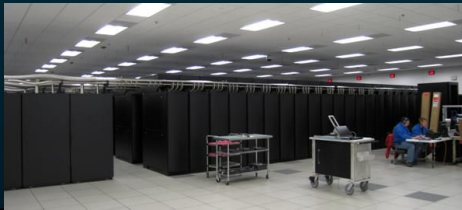
Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

- Inferring source properties from grav. waves *tough*
 - Binary black holes: **7 dims of parameters!**
 - Current NR grav. wave catalogs:
 - ~3,000 waveforms/15 years
(~3 points per dimension)
 - ***Not enough*** moving forward!
 - Need to grow these catalogs!
- Problem: BBH sims need ~4 supercomputing nodes
- Idea: Fit BBH on desktop \Rightarrow massively grow catalogs!



How?

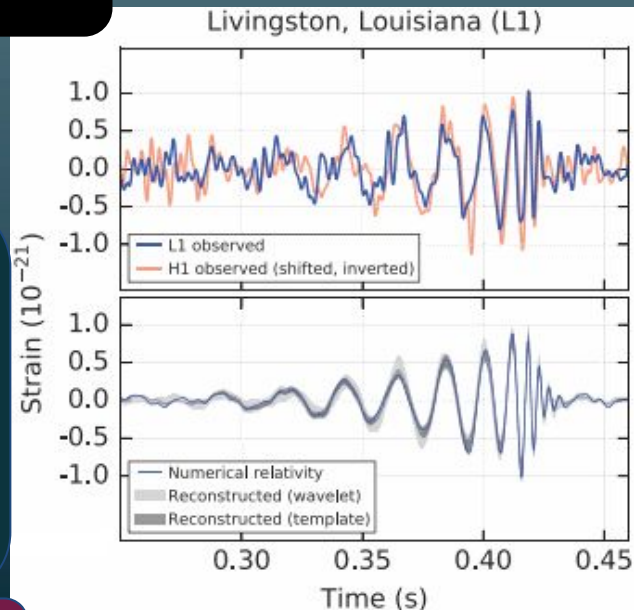


BlackHoles@Home

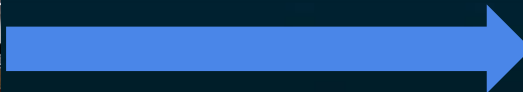
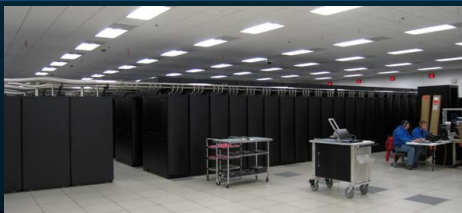
Why do we need it?

Two black holes merge, gravitational waves detected!
Now let's extract the science!

- Inferring source properties from grav. waves *tough*
 - Binary black holes: **7 dims of parameters!**
 - Current NR grav. wave catalogs:
 - ~3,000 waveforms/15 years
(~3 points per dimension)
 - ***Not enough*** moving forward!
 - Need to grow these catalogs!
- Problem: BBH sims need ~4 supercomputing nodes
- Idea: Fit BBH on desktop \Rightarrow massively grow catalogs!



How?



BlackHoles@Home Secret Sauce

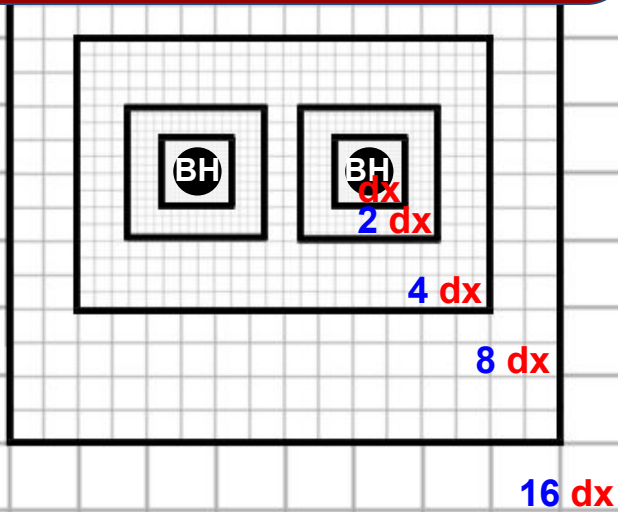
- More efficient numerical grids
- NRPy+

BlackHoles@Home: BBHs on the Desktop

Core Challenge

AMR Grids

*Adaptive Mesh Refinement
(Most Popular Method in NR)*

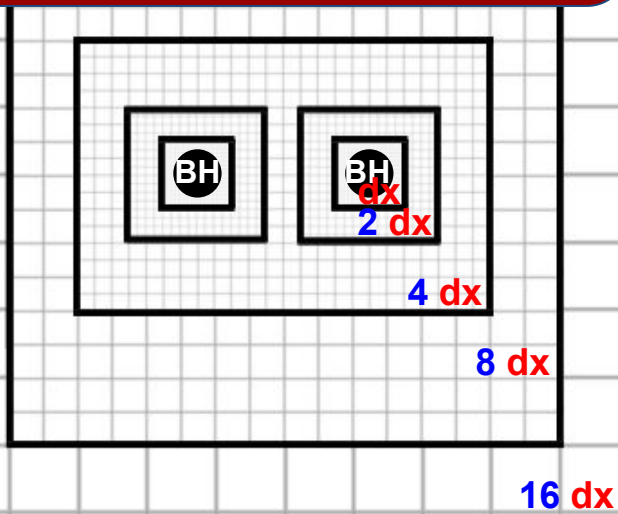


BlackHoles@Home: BBHs on the Desktop

Core Challenge

AMR Grids

*Adaptive Mesh Refinement
(Most Popular Method in NR)*



Better grids would...

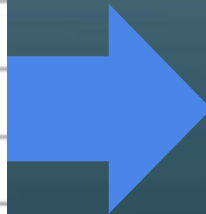
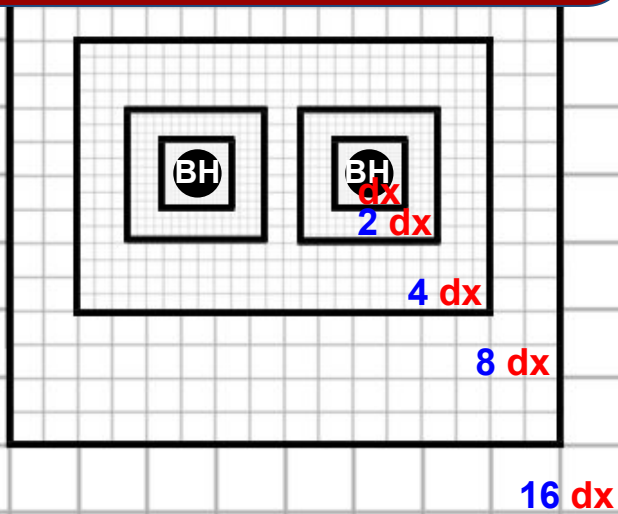
- Exploit near-symmetries
- Employ smoother transitions in resolution

BlackHoles@Home: BBHs on the Desktop

Core Challenge

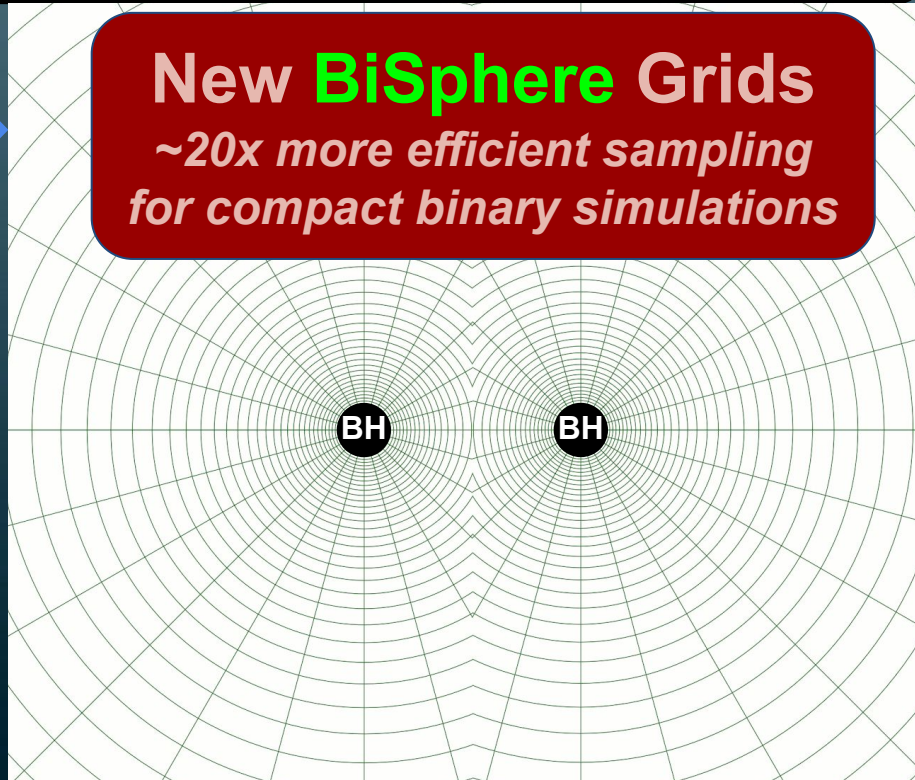
AMR Grids

*Adaptive Mesh Refinement
(Most Popular Method in NR)*



New BiSphere Grids

*~20x more efficient sampling
for compact binary simulations*

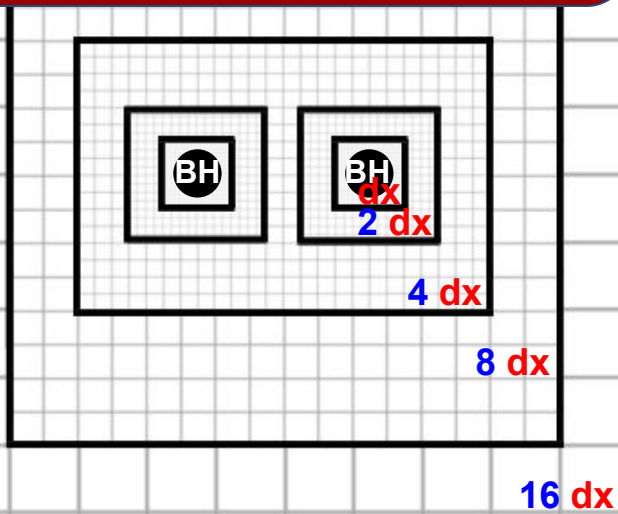


BlackHoles@Home: BBHs on the Desktop

Core Challenge

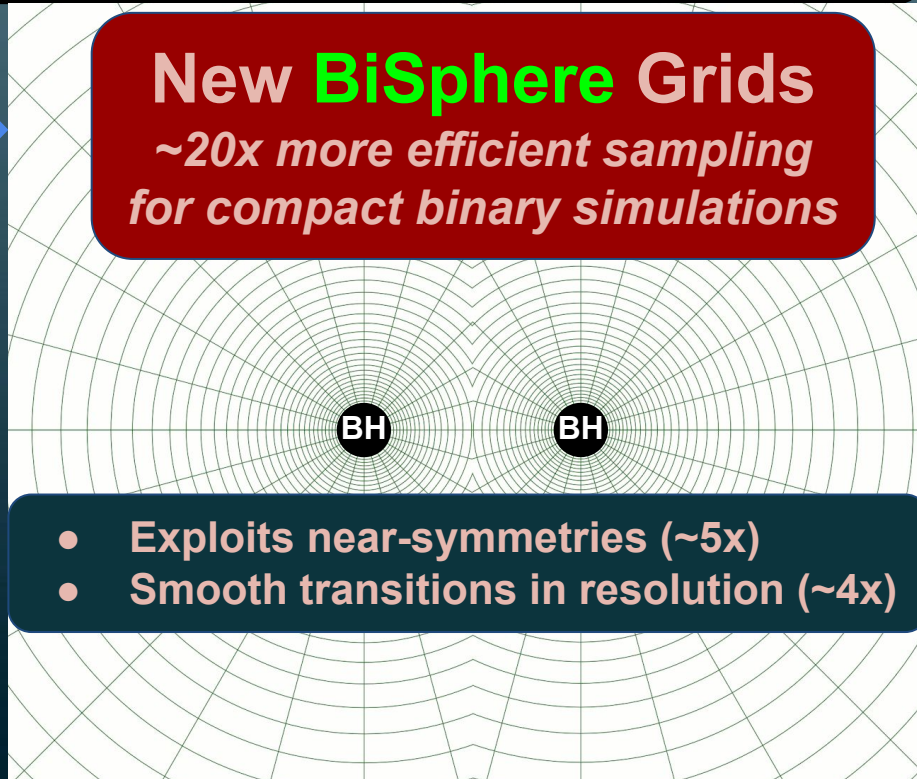
AMR Grids

*Adaptive Mesh Refinement
(Most Popular Method in NR)*



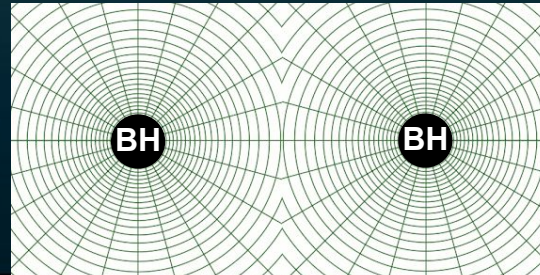
New BiSphere Grids

*~20x more efficient sampling
for compact binary simulations*



BiSphere Challenges

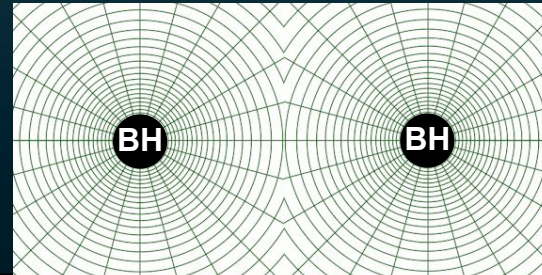
BiSphere grids: Two overlapping numerical grids in Spherical coordinates



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

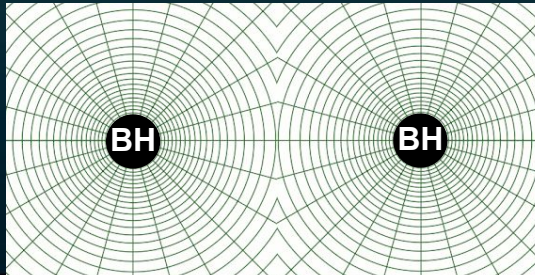
- Challenge #1: Spherical coordinates have coordinate singularities
 - Tensors and vectors diverge or go to zero -> numerically unstable



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in Spherical coordinates

- Challenge #1: Spherical coordinates have coordinate singularities
 - Tensors and vectors diverge or go to zero -> numerically unstable
- Idea: **Scale out singular parts**, (treat singular & nonsingular separately)



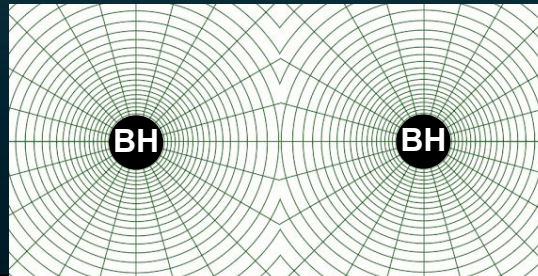
BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in **Spherical** coordinates

- Challenge #1: **Spherical** coordinates have **coordinate singularities**
 - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: ***Scale out singular parts***, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*
 - a. **Ordinary spherical polar: done!**

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)
 - b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)



Need Einstein's equations in Spherical-like coordinates

**NRPy+ : Python-based C code generation
framework for NR**

*Tensorial
expressions in
Einstein-like
notation ⇨
Highly optimized
C-code kernels
(with FDs)*



"Nerpy", the NRPy+ mascot. Photo CC2.0 [Pacific Environment](#) (modified).

 **BlackHoles@Home**

<https://blackholesathome.net>

<https://nrpyplus.net>

<https://github.com/zachetienne/nrpytutorial>

Need Einstein's equations in Spherical-like coordinates

**NRPy+ : Python-based C code generation
framework for NR**

*Tensorial
expressions in
Einstein-like
notation ⇨
Highly optimized
C-code kernels
(with FDs)*



"Nerpy", the NRPy+ mascot. Photo CC2.0 [Pacheco](#) (modified)

 **BlackHoles@Home**

<https://blackholesathome.net>

<https://nrpyplus.net>

<https://github.com/zachetienne/nrpytutorial>



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in **Spherical** coordinates

- Challenge #1: **Spherical** coordinates have **coordinate singularities**
 - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: **Scale out singular parts**, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*

a. **Ordinary spherical polar: done!**

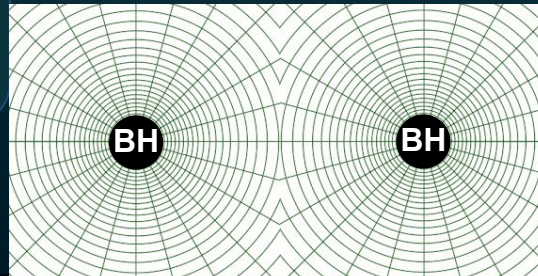
Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)



- Challenge #2: Develop new algorithm for
 - Interpolating between the two spherical grids;
co-orbit grids with the binary system: **done! (early 2019)**
 - Changing basis between the two spherical grids: **done!**



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in **Spherical** coordinates

- Challenge #1: **Spherical** coordinates have **coordinate singularities**
 - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: **Scale out singular parts**, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness on par with Cartesian

a. **Ordinary spherical polar: done!**

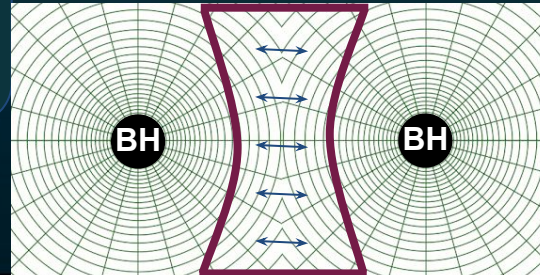
Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)

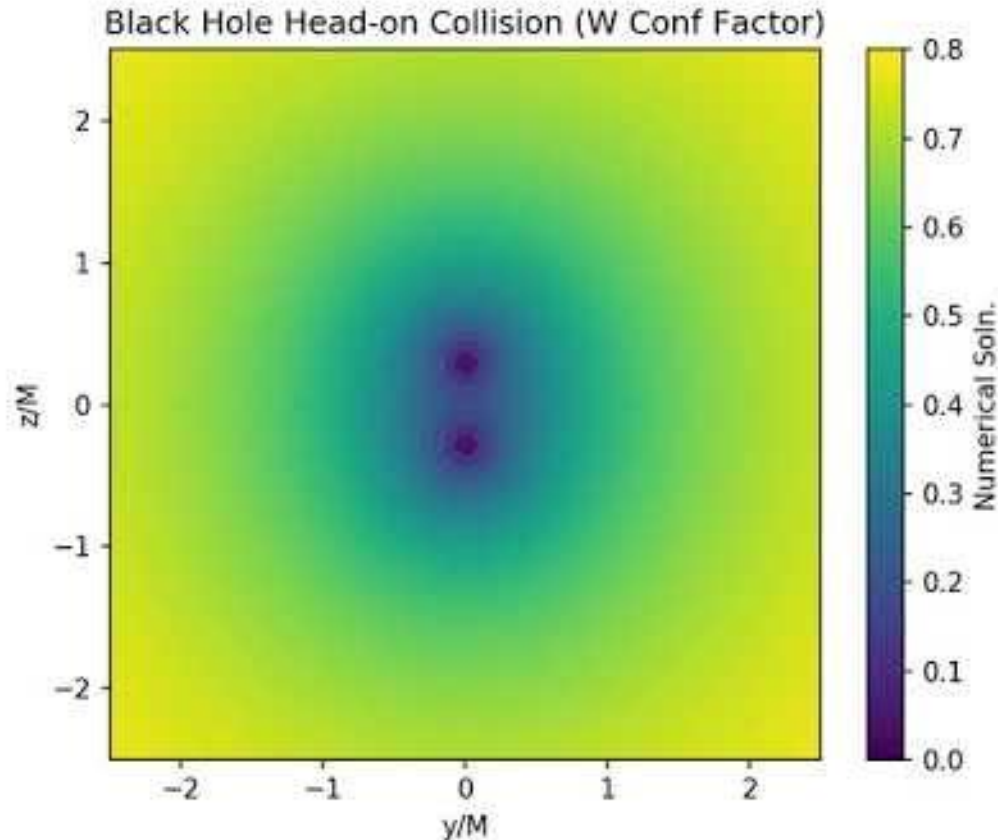


- Challenge #2: Develop new algorithm for
 - Interpolating between the two spherical grids; co-orbit grids with the binary system: **done! (early 2019)**
 - Changing basis between the two spherical grids: **done!**

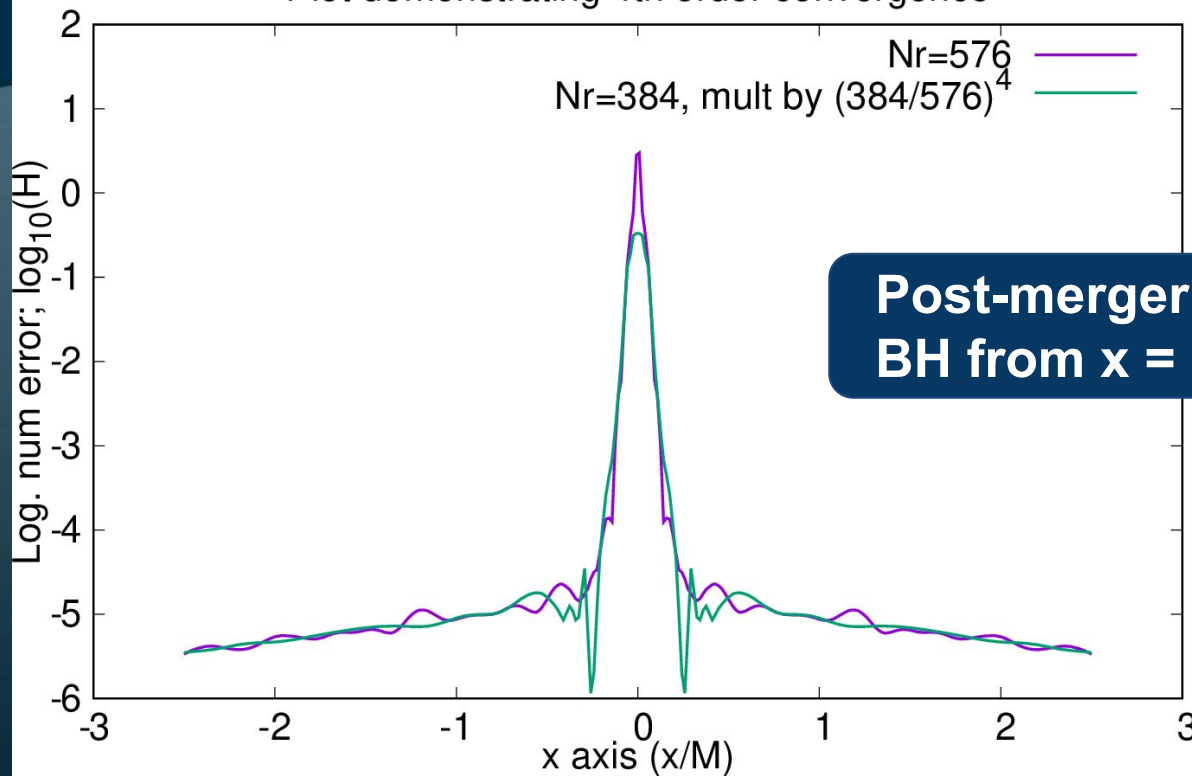


Review: Progress as of 2019 April APS

Link to video (YouTube): <https://youtu.be/ZKPY1nRy-t8>



Plot demonstrating 4th order convergence



**Post-merger num error,
BH from $x = -0.5$ to $+0.5$**

**Finding from BH collision test:
Numerical **errors small** and
converge to zero at expected rate**

BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in **Spherical** coordinates

- Challenge #1: **Spherical** coordinates have **coordinate singularities**
 - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: **Scale out singular parts**, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness on par with Cartesian

a. **Ordinary spherical polar: done!**

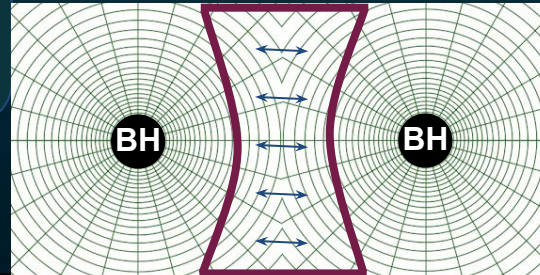
Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)



- Challenge #2: Develop new algorithm for
 - Interpolating between the two spherical grids; co-orbit grids with the binary system: **done! (early 2019)**
 - Changing basis between the two spherical grids: **done!**



BiSphere Challenges

BiSphere grids: Two overlapping numerical grids in **Spherical** coordinates

- Challenge #1: **Spherical** coordinates have **coordinate singularities**
 - Tensors and vectors diverge or go to zero -> numerically **unstable**
- Idea: **Scale out singular parts**, (treat singular & nonsingular separately)
- Result: Numerical stability & robustness *on par with Cartesian*

a. **Ordinary spherical polar: done!**

Baumgarte, Montero, Cordero-Carrión, Müller (PRD 87, 044026, **2012**),
built upon covariant BSSN formulation of Brown (PRD 79, 104029, **2009**)

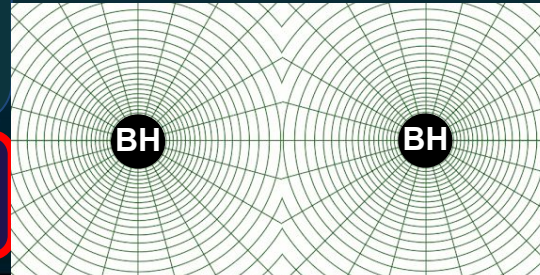
b. **Generic-radius spherical polar (incl. log-radial): done!**

Ruchlin, **Etienne**, Baumgarte (PRD 97, 064036, **2018**)



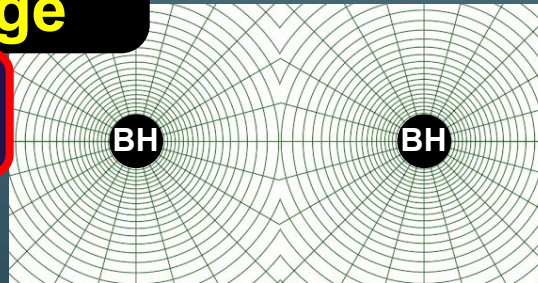
- Challenge #2: Develop new algorithm for
 - Interpolating between the two spherical grids;
co-orbit grids with the binary system: **done! (early 2019)**
 - Changing basis between the two spherical grids: **done!**

- Challenge #3: Simulations **50x too slow!** How to fix?



Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

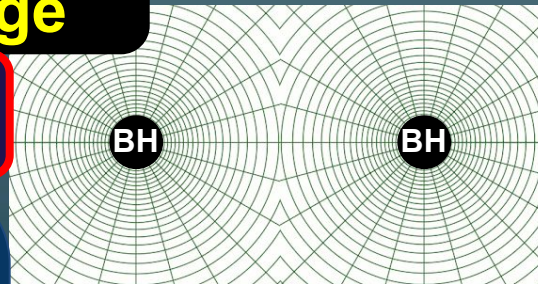


Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!

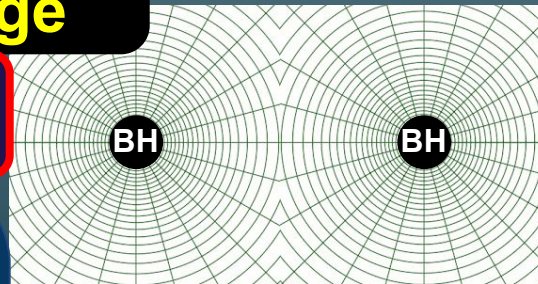


Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis

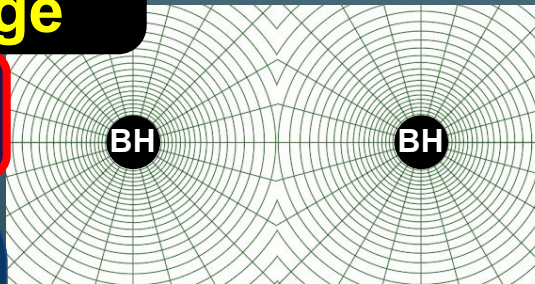


Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis
 - NRPy+ generates NR codes in *many* coord systems
 - Cartesian best for maximizing timestep
 - Spherical-like best for near-axisymmetry

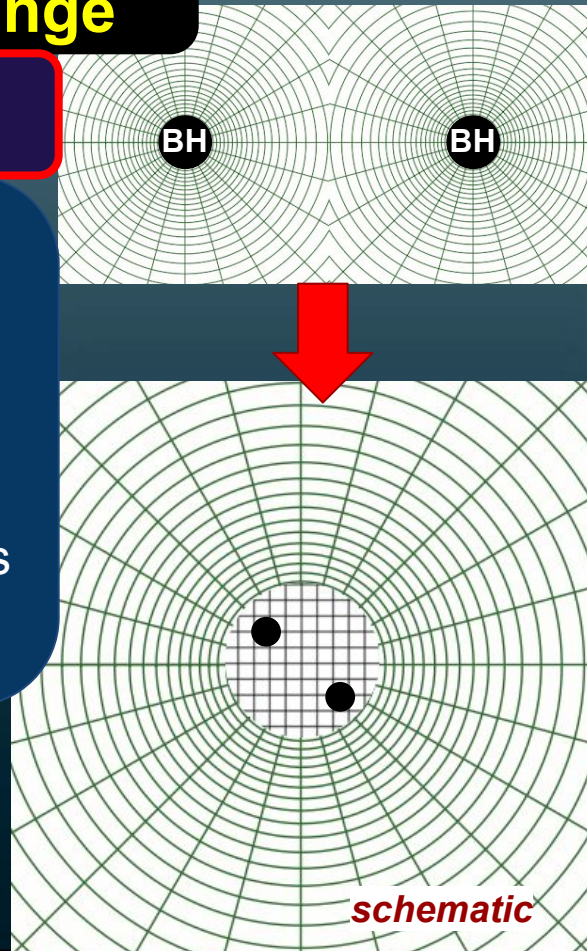


Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis
 - NRPy+ generates NR codes in *many* coord systems
 - Cartesian best for maximizing timestep
 - Spherical-like best for near-axisymmetry



Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

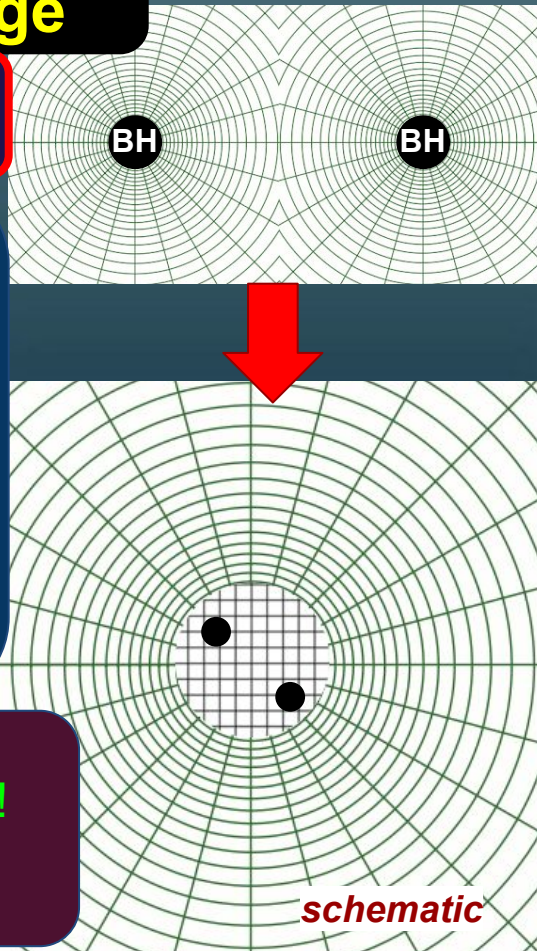
Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis
 - NRPy+ generates NR codes in *many* coord systems
 - Cartesian best for maximizing timestep
 - Spherical-like best for near-axisymmetry



Hybrid gridding approach

- $\sim 100x$ larger timesteps!



Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

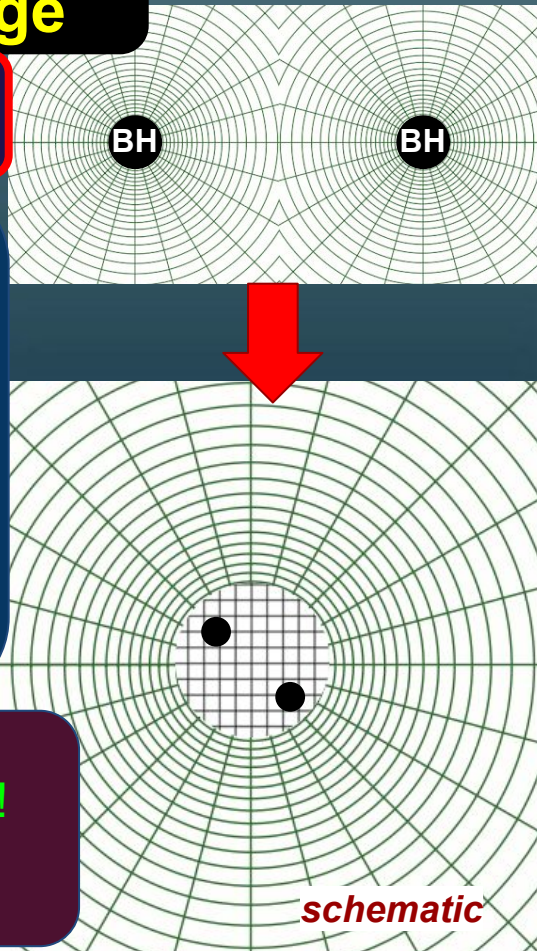
Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis
 - NRPy+ generates NR codes in *many* coord systems
 - Cartesian best for maximizing timestep
 - Spherical-like best for near-axisymmetry



Hybrid gridding approach

- $\sim 100x$ larger timesteps!
- Memory usage?!



Small Timesteps, The Last BiSphere Challenge

- Challenge #3: Simulations 50x too slow! How to fix?

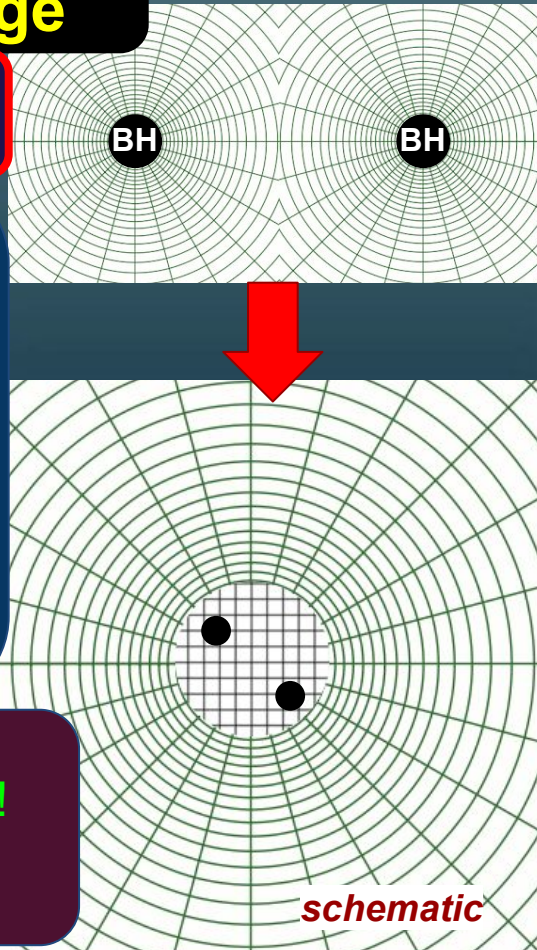
Ideas for making code $>\sim 50x$ faster

- Optimize the codebase
 - $\sim 1.5x$ gain through software optimization!
- Address the source of the problem: CFL condition
 - Simulation timestep \propto min dist between gridpoints
 - Spherical coords focus gridpoints at $r=0$, z-axis
 - NRPy+ generates NR codes in *many* coord systems
 - Cartesian best for maximizing timestep
 - Spherical-like best for near-axisymmetry



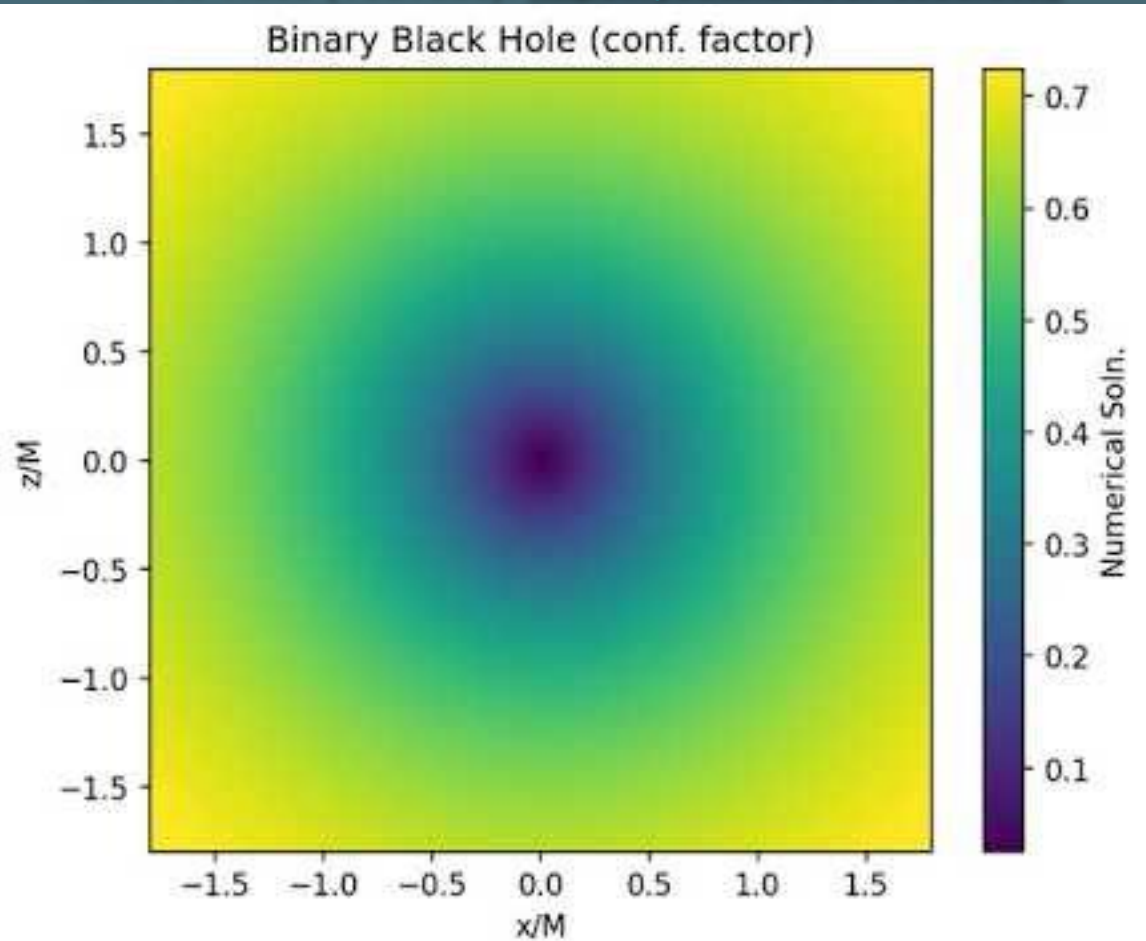
Hybrid gridding approach

- $\sim 100x$ larger timesteps!
- Memory usage?!
- Will it even work?!



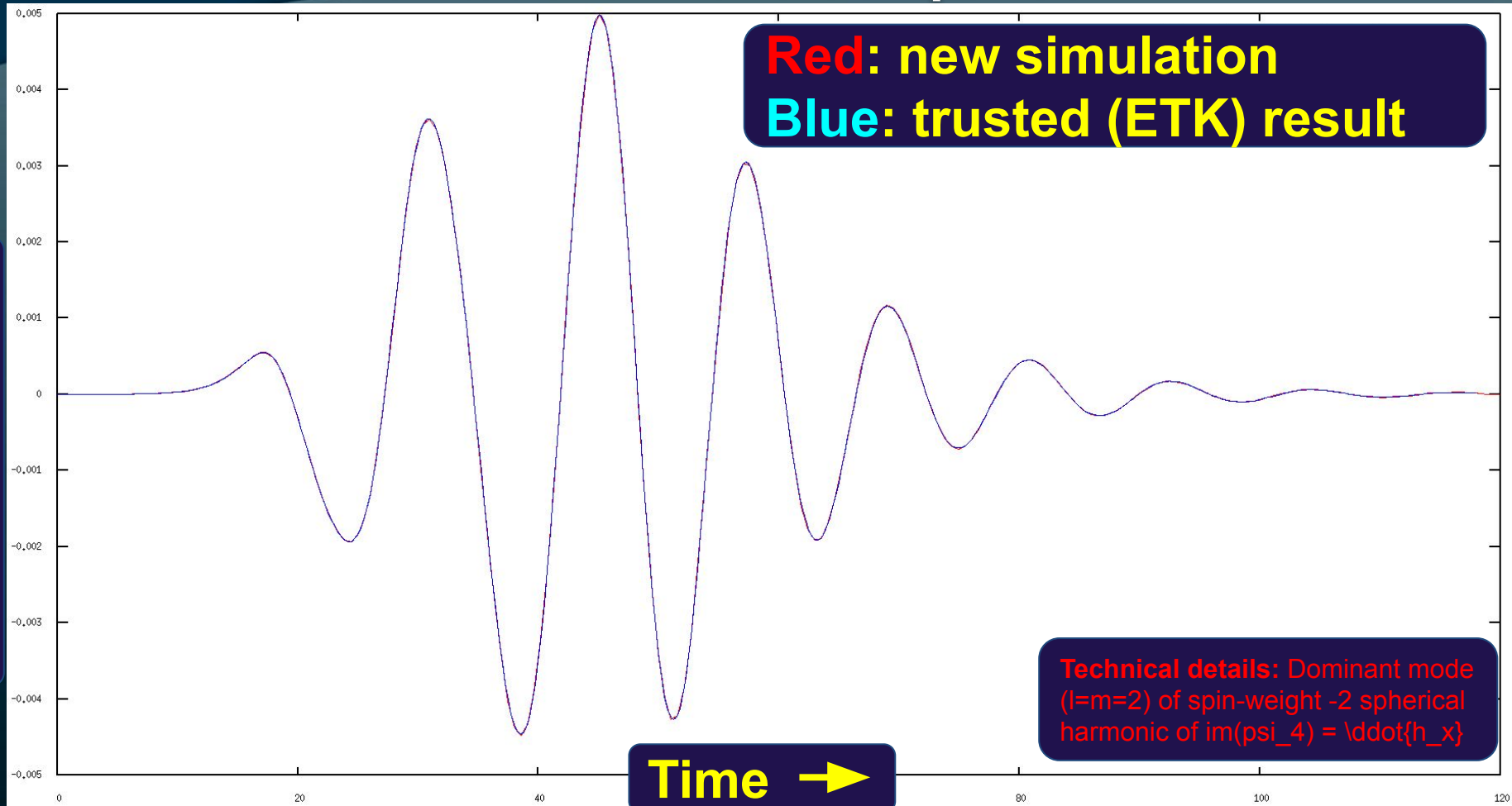
Latest Results

Link to video (YouTube): <https://youtu.be/3CP553oleWc>



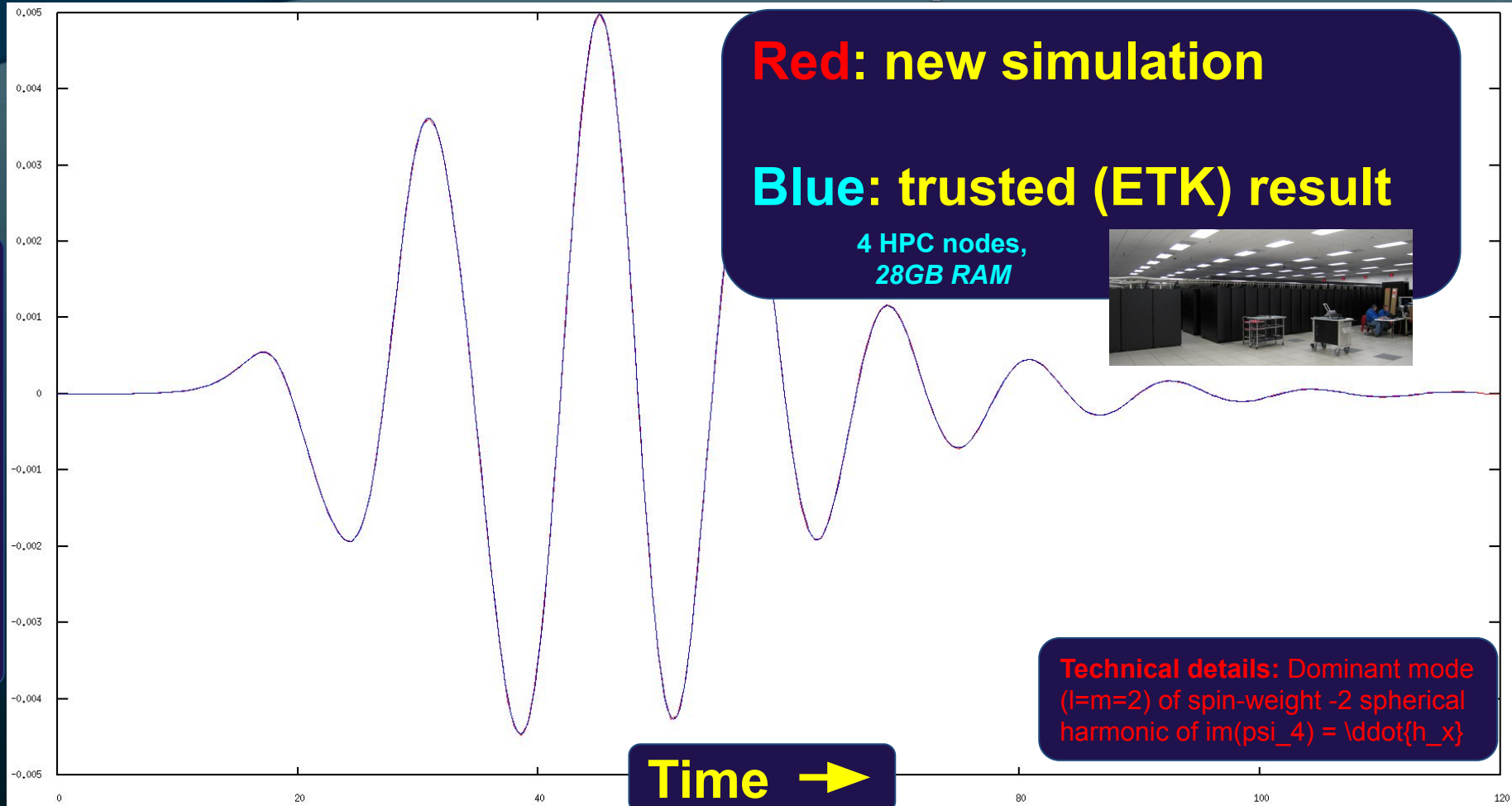
Gravitational Wave Comparison

Wave amplitude



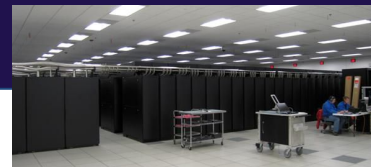
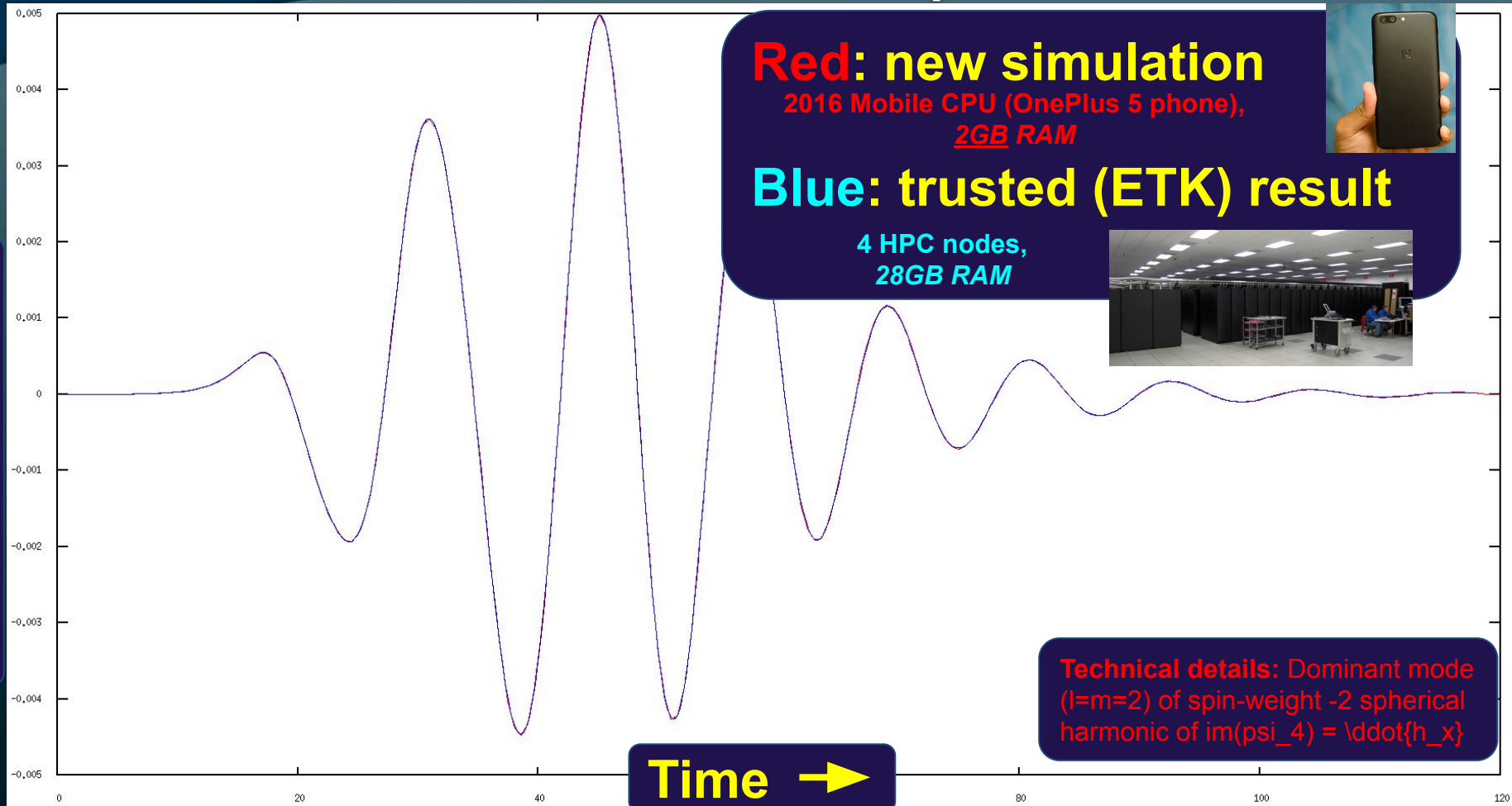
Gravitational Wave Comparison

Wave amplitude



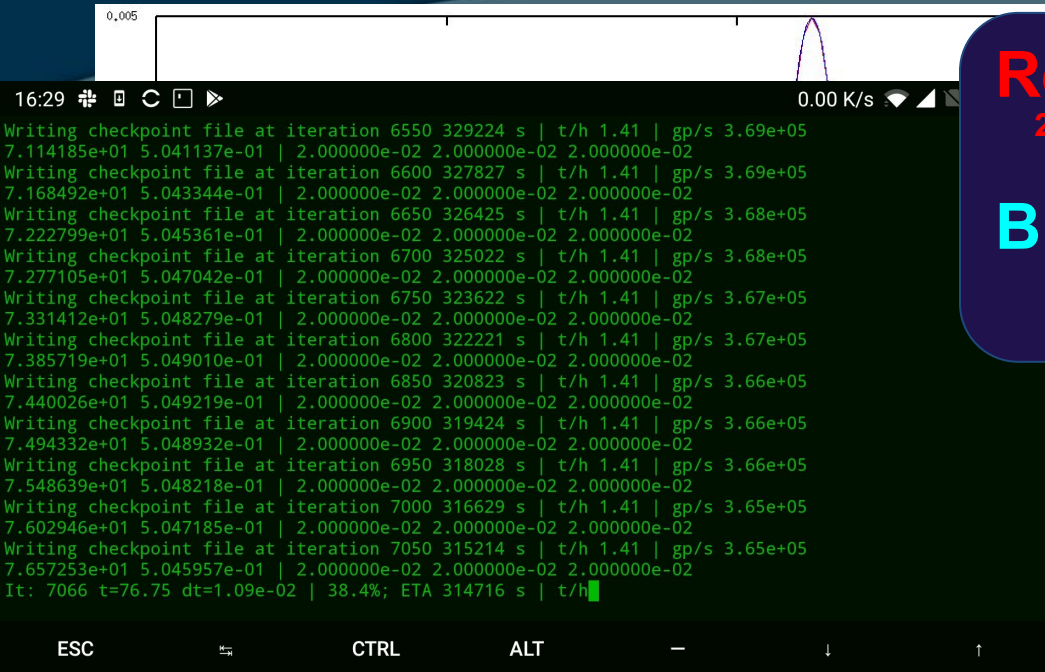
Gravitational Wave Comparison

Wave amplitude



Time →

Gravitational Wave Comparison



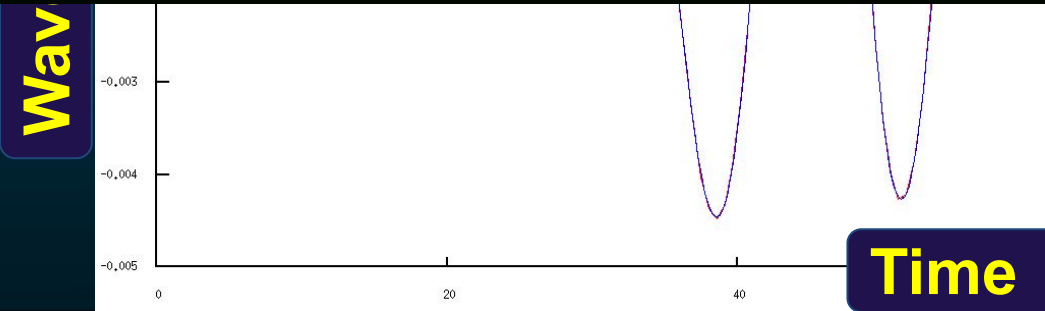
Red: new simulation

2016 Mobile CPU (OnePlus 5 phone),
2GB RAM



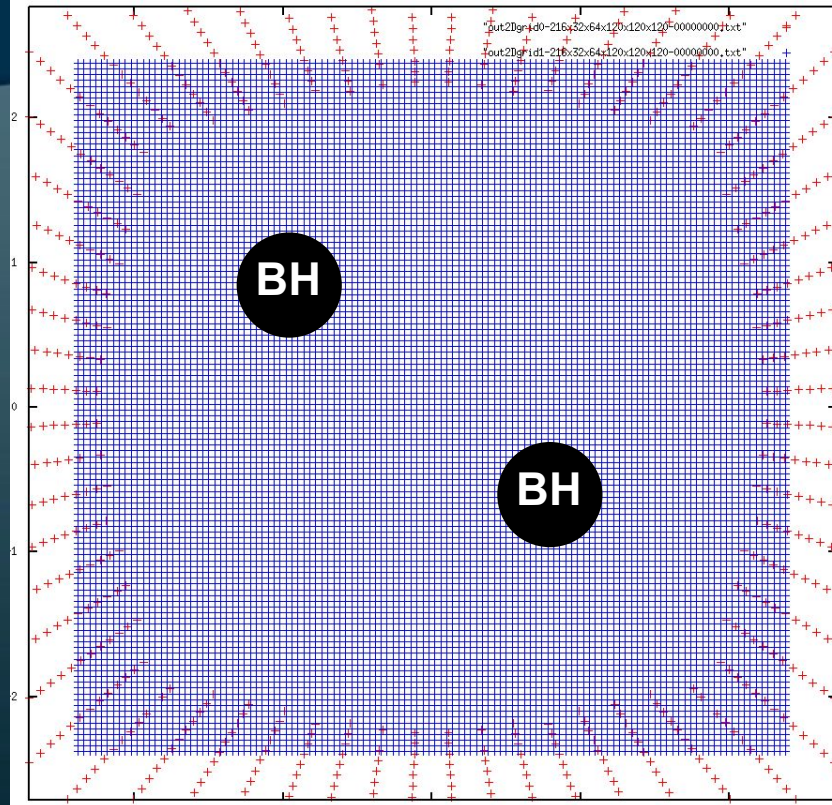
Blue: trusted (ETK) result

4 HPC nodes,
28GB RAM



Technical details: Dominant mode ($l=m=2$) of spin-weight -2 spherical harmonic of $\text{im}(\psi_4) = \ddot{h}_x$

Early 2020 Results



- Problem with this grid structure:
 - Only works well for two orbiting black holes **very close to merger**
 - Larger separations -- Cartesian grid too large -- too much memory!
 - Narrow Cartesian grid & rotate grids? Nope; resolution drop too large
- What to do?!

**Latest:
5-grid Bispheres**

BH

BH

Schematic

**Latest:
5-grid Bispheres**

Cart.

**Small
SinhSpherical**

**Small
SinhSpherical**

Cart.

**Large
SinhSpherical**

Schematic

Latest: 5-grid Bispheres

5-grid Bispheres = Dynamical Grid Structure:

1. Grids corotate with binary
2. Inner four grids migrate inward for inspiral
3. (WIP) Move to simpler grid structure when BHs get really close

Cart.

Small
SinhSpherical

Small
SinhSpherical

Cart.

Large
SinhSpherical

Schematic

Latest: 5-grid Bispheres

Benefits

1. Cartesian grids maximize timestep where Sph grids would focus (no $r = 0$)
2. Large SinhSpherical *excellent* for GW extraction!
3. Low memory footprint

Cart.

Small
SinhSpherical

Small
SinhSpherical

Cart.

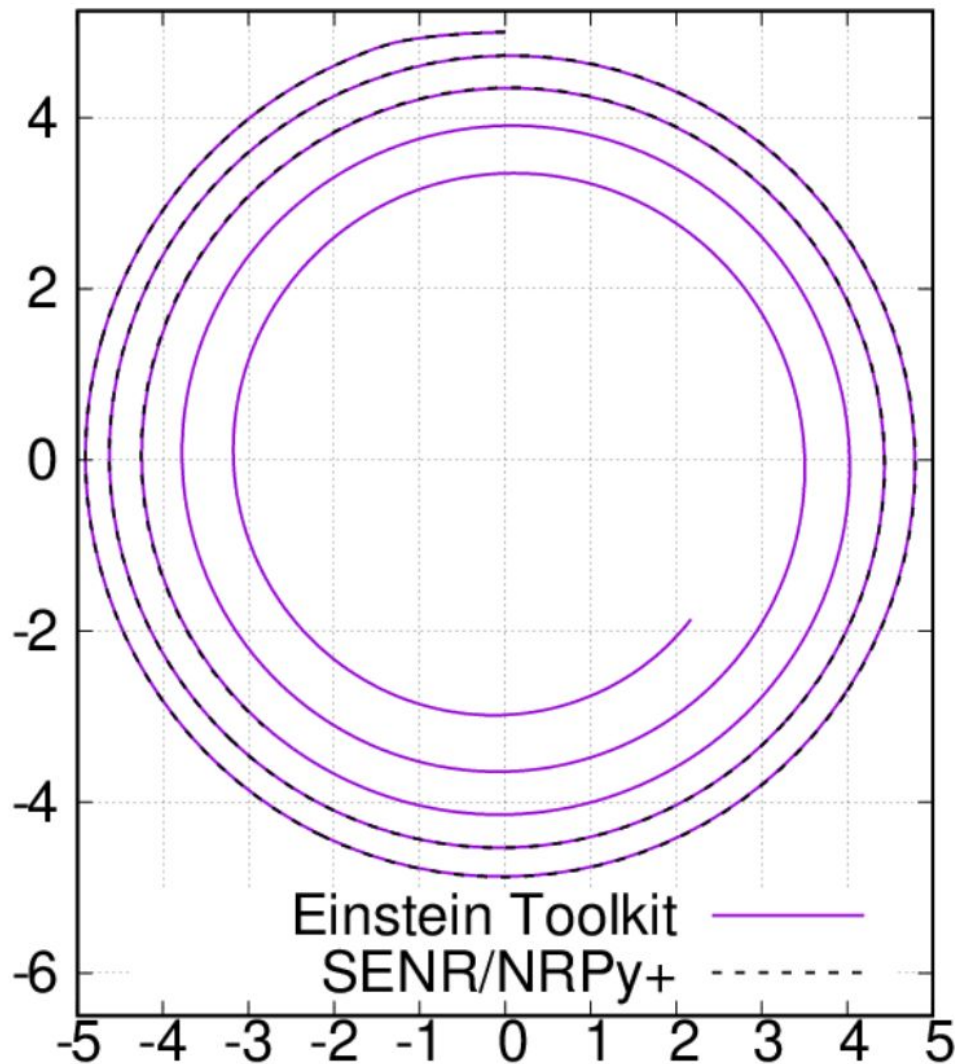
Large
SinhSpherical

Schematic

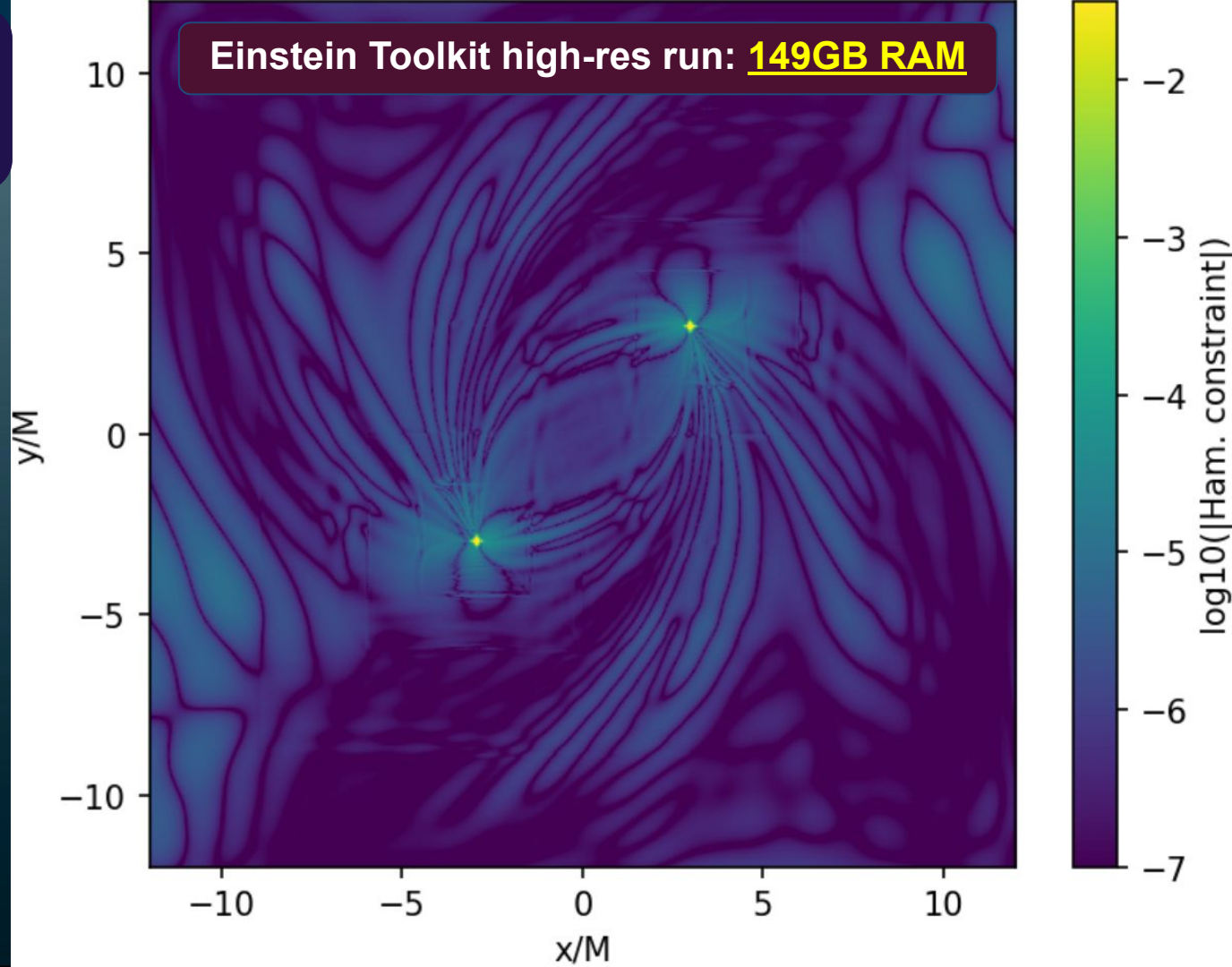
5-grid Bispheres: Binary Trajectory Comparison

Binary Trajectory

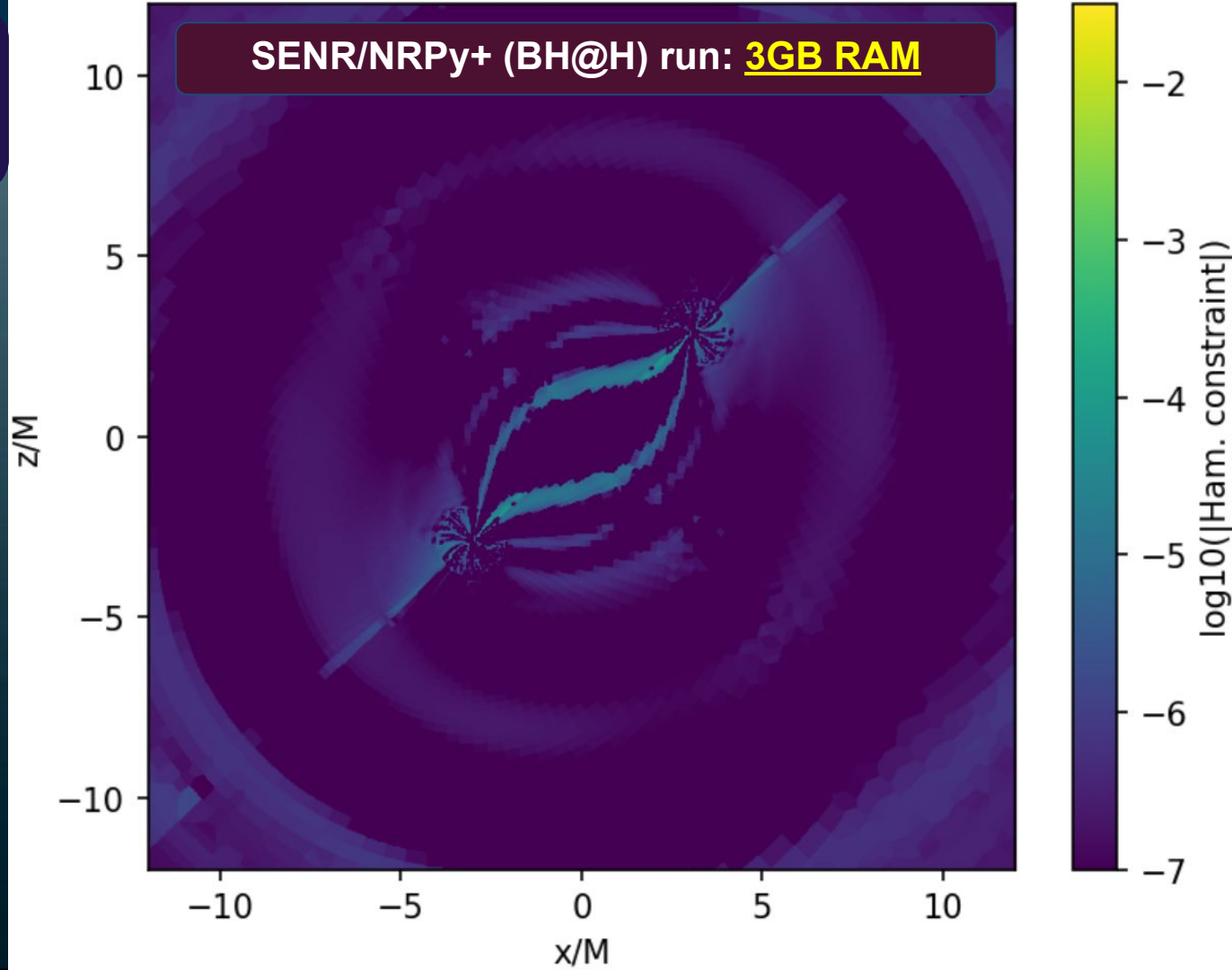
1. Same GR gauge condition -> trajectories should overlap (*They do!*)
2. Einstein Toolkit high-res run:
149GB RAM
3. SENR/NRPy+
(BlackHoles@Home):
3GB RAM



5-grid Bispheres: Numerical Error Comparison



5-grid Bispheres: Numerical Error Comparison



Conclusions

Summary

1. (close-separation) BBH mergers on a cellphone!
2. 5-grid Bispheres: Comparable or superior numerical errors vs Cartesian AMR
~50x less memory usage!

BH

BH



Conclusions

Summary

1. (close-separation) BBH mergers on a cellphone!
2. 5-grid Bispheres: Comparable or superior numerical errors vs Cartesian AMR
~50x less memory usage!

BH

BH

We're so close now!

-={ Last Steps }=-

1. Current focus: GW extraction
2. Next:
 - Migrate to simpler gridding structure at merger
 - Develop & release the kraken BOINC client!



Conclusions

Summary

1. (close-separation) BBH mergers on a cellphone!
2. 5-grid Bispheres: Comparable or superior numerical errors vs Cartesian AMR
~50x less memory usage!

BH

We're so close now!

-={ Last Steps }=-

1. Current focus: GW extraction
2. Next:
 - Migrate to simpler gridding structure at merger
 - Develop & release the kraken BOINC client!

BH

Stay tuned! BlackHoles@Home will be launching soon

<https://blackholesathome.net>

Schematic

